An Image Schema Language

Robert St. Amant[1], **Clayton T. Morrison**[2], **Yu-Han Chang**[2],
Wei Mu[1], **Paul R. Cohen**[2], and **Carole Beal**[2]
(stamant@ncsu.edu, cohen@isi.edu)

[1] Department of Computer Science
North Carolina State University
Raleigh, NC, 27695

[2] Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292

Abstract

This paper introduces ISL, a language for representing and manipulating image schemas. ISL supports the representation of symbolic as well as quantitative dynamic properties of objects and relationships. We have encoded a number of the image schemas commonly covered in the cognitive linguistics literature and tested them in three domains: patterns in chess, tactics in military scenarios, and behavior in a simple robot arm simulation. This paper discusses the design of the language and demonstrates its representational capabilities with examples from these domains.

Introduction

In cognitive linguistics (Oakley 2006), "an image schema is a condensed redescription of perceptual experience for the purpose of mapping spatial structure onto conceptual structure. According to Johnson (1987), these patterns 'emerge as meaningful structures for us chiefly at the level of our bodily movements through space, our manipulations of objects, and our perceptual interaction'." Another source (LinguaLinks 2003) has it that "an image schema is a mental pattern that recurrently provides structured understanding of various experiences, and is available for use in metaphor as a source domain to provide an understanding of yet other experiences." Image schemas have also been suggested to play a critical developmental role, forming the basis of early cognitive development, and possibly extending to all sensori-motor perceptual modalities (Mandler 1992, 2004).

For over two decades, cognitive linguists have developed accounts of how the semantics of words and sentences can be explained in terms of image-schematic representations (e.g., Lakoff 1987, Gibbs & Colston 1995, Talmy 2003). Some words have unadulterated physical meanings, but many transfer the original physical meaning to non-physical situations. As you grasp this point, you grasp it in a nonphysical way, yet a large chunk of the original physical meaning of grasp remains.

Although the theory of image schemas accounts for lexical semantics pretty well, almost all accounts are post-hoc. Some steps have been taken toward the computational formalization of image schemas (notably, Bailey 1995 and Regier 1996), but image schemas are still largely discussed in qualitative, abstract terms.

In this paper we introduce ISL, a language in which image schemas can be modeled computationally. ISL has been under development for almost a year. We have applied it to three domains: patterns in chess, tactics in military scenarios, and behavior in a simple robot simulation. As we developed ISL we also learned important lessons about image schemas. This paper touches on four: First, there is an inherent ambiguity in accounts of schemas like "path"—it is unclear whether it means "a physical configuration" or "the path I intend to follow." Since we want our image schemas to serve an intentional agent, this ambiguity had to be resolved. Consequently, we distinguish three kinds of schema: static, dynamic, and action. Second, image schemas for verb-like concepts need several parts: controllers, "maps" of dynamic behavior, role bindings, and associated axioms. Third, the previous two points drive home the idea that many image schemas require quantitative and procedural components as well as a symbolic/declarative component. The difference between "brushing", "bumping", and "crashing" into a wall, for example, depends on quantitative rather than symbolic properties of the interaction, yet we also need to bind entities and declaratively represent relations that we track over time. Finally, implementing image schemas has given us insight about how they can function as a semantic core for reasoning.

Image Schemas & Cognitive Architecture

Image schemas are integrally tied to perception and motor function, but serve as the bridge to higher-level cognition. Figure 1 locates image schemas in a simple schematic of a cognitive architecture, cutting across the boundaries of low to high level perceptual-motor function, to the basis of higher level cognition where we find deliberative reasoning, planning and problem solving. We believe image schemas serve to organize and represent characteristic perceptual-motor patterns to form a *semantic core* on which higher-level cognition rests.[1]

A full computational model of image schemas will include aspects of the mapping from lower-level to higher-level perception. There is ample evidence that we reuse our perceptual and motor systems for modeling and reasoning, such as mental imagery (Kosslyn 1994), and surely this plays a role in the representational power of image schemas. But image schemas also have distinctly compositional and symbolic properties. In this paper we focus on the upper half of the image schema

---

[1] This view is consonant with Barsalou's (1999) proposal that the semantic core is based on a *perceptual symbol system*.

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **An Image Schema Language**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **North Carolina State University,Department of Computer Science,Box 7534,Raleigh,NC,27695-7534**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES **The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

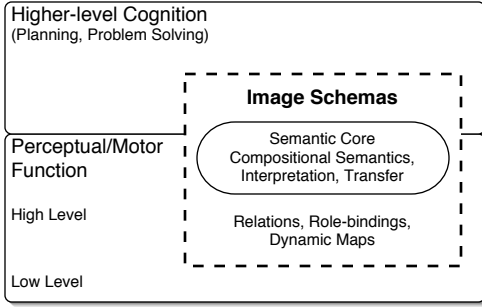| 16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **6**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Figure 1: Locating schemas in a cognitive architecture.

portion of the architecture, the logical structuring of image schemas that forms the basis of the semantic core (depicted as the oval in Figure 1). This sets our ideal target for ISL: at this level, image schemas have a relational structure with compositional semantics that admits operations of interpretation and permits cross domain transfer of image schema structure. ISL should represent image schemas as general-purpose syntactic forms (representations) with the property that syntactic operations on them are equivalent to semantic operations in an indefinitely large number of domains.

Properties of ISL

Informing our representation is a catalog of linguistically-derived image schemas provided by Croft and Cruse (2004), reproduced (with some modifications) in Table 1. In their most basic form, these image schemas can be taken as abstract descriptions of objects and relationships. For example, a containment relationship exists between its contents and a container consisting of an inside, an outside, and a boundary. A path consists of a starting location, an end location, and a (possibly continuous) set of intermediate points.

Space:	Location, Up-Down, Front-Back, Left-Right, Near-Far, Verticality, Center-Periphery, Straight, Contact
Force:	Compulsion, Blockage, Diversion, Counterforce, Restraint, Resistance, Attraction, Enablement
Containment:	Container, In-Out, Surface, Content, Full-Empty
Locomotion:	Momentum, Path
Balance:	Axis Balance, Twin-Pan Balance, Point Balance, Equilibrium
Identity:	Matching, Superimposition
Multiplicity:	Merging, Collection, Splitting, Iteration, Part-Whole, Linkage, Count-Mass
Existence:	Removal, Bounded space, Cycle, Object, Process, Agent

Table 1: Image schemas

As represented in ISL, image schemas are objects, as in the object-oriented data model.[2] Each schema has a set of operations that determine its capabilities. For example, operations for a basic container schema include putting material into a container and taking material out. Each schema also has a set of internal slots that function as the equivalent of roles in a case grammar sense (Fillmore, 1968). Slots permit image schemas to be related to each other through their slot values. For example, the contents of a container can be other image schemas; containers are one way that we intuitively understand the concept of sets.

An important aspect of ISL is its use of *interpretation*. In object-oriented terms, interpretation can be thought of as an extended form of delegation. Interpretations map from one or more specifications of a "source" image schema to a "target" schema. For example, we would probably first think to represent a room as a location or bounded space (i.e. a region) image schema, but from a fire marshall's perspective it would be useful to interpret a room as a container with a capacity of some number of people. Interpretation gives us flexibility in evaluating the properties of some domain in terms of image schemas; different (even conflicting) interpretations can be maintained at the same time for a single "real" object or relationship. Interpretation is also critical to metaphorical extension and bears relations to analogical mapping (Gentner & Markman 1997).

To illustrate the use of ISL, it will be helpful to walk through an example, which we take from our work on representing chess patterns. Consider a chess board in which the Black queen has the White king in check. In image schema terms, we say that there exists a path from the queen to the king. In ISL, we generate a path schema, which contains a set of locations, as shown in Figure 2. Representing a path simply as a set of locations gives us generality, but here it's important that the queen can traverse the path in the situation that holds currently on the board. This is captured by an interpretation of the path as a set of directional linkages from each location (a source) to the next on the path (a destination). Another piece of domain information is that no location can be occupied by more than one piece at a time. This is represented by an interpretation of each location as a container with a capacity of 1. When a piece moves to a location, the container reaches capacity and yet another image schema, empty/full, is automatically created, indicating that the location is full.

Given these image schemas, their relationships, and the operations that they support, it becomes possible to reason about the situation and the possible responses White can make to counter the threat of the queen. The check exists because the path from the queen to the king is traversable. Traversability for a path schema is defined, in words, as follows: a path can be traversed when every linkage between successive locations can be traversed. Traversability for a linkage schema, in turn, is allowed when its source can be entered and its desti-

---

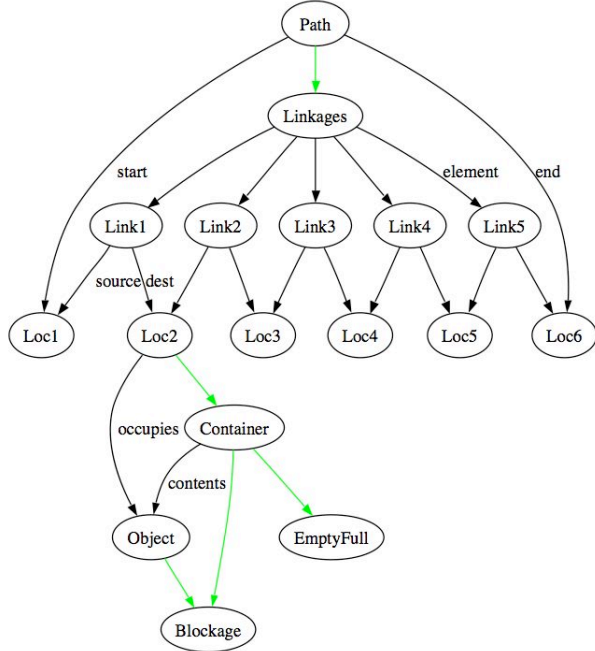[2]...which in turn is descended from the frame knowledge representation model (Fikes & Kehler 1985).

Figure 2: Representing blockage in ISL.

nation can be exited. Basic locations have no built-in constraints on entering and exiting, but when a location is interpretable as a container, this changes. One cannot add more to a container that has reached capacity. The interpretation relationships between these schemas cause changes to propagate outward: a full container cannot be added to; its location cannot be entered; a directional linkage cannot be traversed (via its source); a path cannot be traversed (due to a non-traversable linkage). The result is a new image schema, blockage, which is created when a container representing a location that acts as the source of a directional linkage in a path becomes full. The contents of the container constitute the blocker. This structured combination of image schemas—locations, path, linkages, blockage, and so forth—can be stored away in memory for later retrieval, limiting the need for a complete reconstruction of the combination from scratch.

The ISL representation provides a description of the situation in the form of a structured combination of image schemas. Compare this combination with how we might describe a tactic in chess: "When an opponent's piece puts your king in check, you can counter by moving another piece into its path." The combination of schemas captures the essence of this natural language description. The representation is general, abstracting away the specific positions of the pieces, the existence of other pieces, even the identity of the attacking piece. The generality of the representation can also be seen in that its substructure maps to other basic concepts in chess. By using object schemas that include information about the color of a piece, we can use the path/linkage substructure to represent a threat of one piece

on another, when the colors of the pieces are different; if they are the same, we can represent a defense relationship. The representation also supports the ability to reason about emergent structure. White might have a dozen possible moves in the situation given in the example, but few of them will be appropriate. One of White's most plausible responses, in terms of image schemas, is to recognize that the situation is a partial match to a blockage schema (which does not yet exist), and that a specific response will lead to the creation of the blockage. Rather than reasoning about the low-level properties of individual pieces, White reasons using tactical abstractions. Other chess concepts similarly lend themselves to abstraction that can be naturally captured by image schemas: application of force on the opponent's king (even if the king is never put in check), balance in the distribution of pieces on the board, control of the center of the board, and so forth. Lower-level descriptions of moves (e.g., based on paths alone) are not inaccurate, but they fail to capture the reasons behind the moves.

Types of ISL Image Schemas

The symbolic representation provided by ISL can capture a variety of chess patterns, but other domains lack the representational simplicity of chess. For example, in some physical environments, properties vary over continuous ranges; time marches forward rather than stopping for turn-taking; descriptions hold to a greater or lesser extent. If ISL were limited to symbol manipulation, it would fall prey to many of the same problems faced by early attempts in AI research to capture realistic environments (e.g. Schank & Abelson 1977).

To address these issues in ISL, we distinguish three general types of image schemas. In the following sections we describe these image schema types in more detail.

Static Schemas

Static schemas are instantaneous descriptions of non-process relationships. The examples in the previous section give a reasonable overview of static schemas, but for contrast with dynamic and action schemas it will be helpful to see how static schemas are created and combined. Consider an agent A in some environment with a ball B. A's sensory input includes its distance from B, which allows A to generate a static near-far schema for the non-commutative relationship $\{A, B\}$. The slots of the near-far schema include this distance and the (domain-dependent) degree to which B is near to or far from A. For simplicity, we might say that if A can come into contact with B without changing its location (e.g., by reaching rather than walking), then B is near A to a high degree. In cases where sufficient domain information is not available, the degree slot can be left empty.

The near-far schema is created automatically based on the input from A's sensors. Other schemas can be generated as interpretations of the near-far schema. For example, if A and B are so close that they are essentially in the same place, then a superposition or a contact schema can be generated as an interpretation of the near-far relationship. In ISL, this is expressed in a

declarative form using ISL constructs that act as production rules. When the predicates associated with a specific relationship hold, based on the information provided by the near-far schema, a superposition schema is generated and attached as an interpretation of the relationship.

Importantly, static schemas represent instantaneous relationships at any point at time. They become active and change when perceived conditions change. But they do not represent change itself. To incorporate dynamics, a second type of schema is used.

Dynamic Schemas

While a static schema is adequate to represent a snapshot in time of the relationship $\{A, B\}$, there are many cases where we must also represent the dynamics of a relationship. For example, A may be far from B but moving in B's direction, a dynamic situation that we can naturally capture as an approaching schema. The approaching schema is a dynamic extension of the static near-far schema. In order to identify relationship dynamics, dynamic schemas are associated with *recognizers*. The recognizer for an approaching schema tests the distance between A and B at time intervals (or rather the slot of the near-far schema representing the relationship), to determine whether the distance is decreasing. When this is the case, an approaching schema is generated; when not, any existing approaching schema for the relationship is destroyed.

At any point in time, a large number of dynamic and static schemas may be active. Some schemas, for example the approaching relationship, may appear and disappear (e.g., consider the relationship between you and the car in front of you as you slowly move through stop and go traffic). The properties of a given schema may change over time as well. All of the schemas and their properties, taken together, constitute the state of the environment. Of course, not all of this information is relevant. For example, while A is approaching B, A is also approaching other objects and locations that happen to be near B. Determining what is relevant is the subject of future research on mechanisms for focus of attention. For now we constrain dynamic relations generated.

So far we have not said how dynamics—changes in state variables over time—are represented in ISL. Expanding on previous work using *dynamic maps* to represent dynamics described by verbs used by children and adults (Cohen 1998; Cohen, Morrison & Cannon 2005), ISL uses dynamic maps to represent continuous state changes. In this case, a map is a space whose dimensions correspond to variables, such as distance, relative velocity or energy transfer. Changes in state variables tracked over time are then represented as trajectories through the map space. Characteristic regions or trajectories (directed paths through regions) can be used to describe classes of dynamics. For example a map for approach for the $\{A, B\}$ relationship records trajectories of the decreasing relative distance between A and B over time. In the same way, the recognizer for the approaching schema tests for decreasing distance trajectories over a short interval of time, and if the observed trajectory matches a prototypical decrease, the approaching dynamic schema is created; if the trajectory later diverges from that prototype, the approaching schema is removed.

Action Schemas

Even with the added representational flexibility of dynamic schemas, ISL is still missing an important property in its description of A: A is an agent with intentions. A can choose to take some actions rather than others, and these lead to different behaviors and outcomes in the environment. ISL thus includes a representation of action schemas, each associated with a controller.

In the example of approaching given above, if A is to approach B, an approach action schema is selected. Its controller determines an appropriate action or sequence of actions to take in order to reduce the distance between A and B. In this arrangement, the dynamic schema that recognizes "approaching" acts as an expectation for the result of taking the action.

Individual dynamic schemas are sufficient to specify very simple behaviors resulting from actions, but their scope is limited. Consider the approach action schema above: eventually A reaches B, and approaching is no longer relevant. A qualitative change occurs, which can be represented by the appearance of a contact schema (and, if the agent is moving with sufficient speed and the ball is light enough, a movement schema is created and associated with the ball as it is pushed away).

To represent these transitions between dynamic states, we use the formalism of state machines. States captured by static and dynamic schemas, as discussed in the previous two sections, can be chained together with actions taken by the agent. For example, we might represent the agent A "kicking" the ball B as a sequence of three dynamic states: A approaching B, A coming in contact with B, and A stopped with B moving away (Cohen 1998). Of course, there may be possible transition to different states, depending on the parameters of controllers or other conditions. For example, if A's velocity decreases to zero at point of contact, then the expected transition to B moving away may not happen—no energy is transferred to B and the two remain in contact. While static and dynamic schemas are typically associated with individual states in a state machine, action schemas may include transitions between multiple states.

A Continuous, Dynamic Example

A simple physics simulation illustrates using ISL with all three types of image schemas to describe a dynamic environment. The domain is a simple "playpen" environment with ballistic physics, modeled in the breve 3-D simulation engine (Klein 2002). Figure 3 shows a snapshot of the playpen with a "cat" (red ball) and an "agent" (blue rectangle) in an open field surrounded by the walls of the playpen. The cat is programmed to run away from the agent if the agent gets too close or approaches too fast. When the agent is a reasonable distance away and not moving too fast in the cat's di-
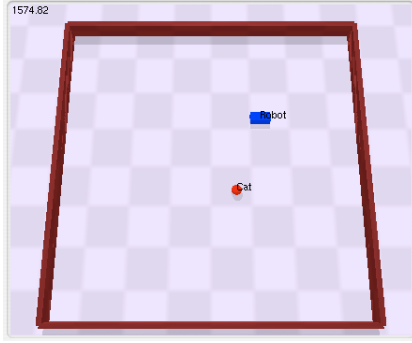
Figure 3: A simple toy domain in breve.

Figure 4: A schema-based state machine describing the agent's possible interactions with either a ball or a cat.

rection, the cat is not "threatened." Given this behavior, an effective way for the agent to catch the cat is to move to the cat slowly (sneak), and then move very rapidly to the cat (pounce) once the cat is close.

Using ISL, we can build a state machine that completely describes the agent's potential interactions with the cat. Each state is described by sets of image schema instances. For example, instances of the static near-far schema maintain information from the environment about the distance between the agent and other objects, such as balls, cats, and walls. In its interaction with balls, over a large number of scenarios, the agent finds that it can come into contact with a ball simply by applying any controller (via an action schema) that it has available for approaching. For cats, however, the situation is different: the cat's behavior depends on its distance from the agent. An approach-slowly controller is appropriate for sneaking up on the cat at a distance, while an approach-fast controller is appropriate for the pouncing phase. The differing outcomes in the environment arising from the cat's behavior at different distances (as well as the actions that turn out to be successful at different distances) give rise to a natural distinction between near and far in the near-far schema. Below some threshold (in this case $distance \leq 6$), the cat is near the agent; above that threshold the cat is far from the agent.

Figure 4 shows the state machine that describes the agent's interactions with balls and cats in the Breve simulation. The state S2, for example, says that if the cat is near the agent, and the cat's velocity is slow, then if the agent can execute the fast-approach action schema it should be able to make contact with (i.e. catch) the cat.

State machines like this one play several useful roles for an agent. First, the agent can use the state machine to formulate plans, in this case for catching the cat quickly, by identifying desirable transitions between states. Second, the state machine provides a general description of such plans, once numerical values have been abstracted away. For example, "I should first move slowly toward the cat, then faster," regardless of specific values for "slow" and "fast". Third, if the state machine has been constructed appropriately, the agent can
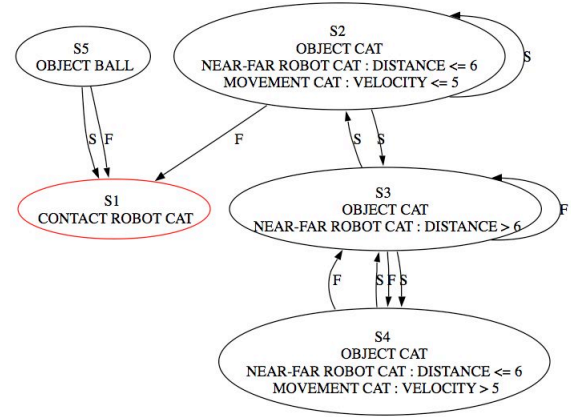
in principle identify what properties of the environment lead to its success or failure in some task. For example, at the most abstract level, "I was not able to contact the cat because it did not behave like a ball," or, more specifically, "When I approached the cat, it moved farther away."

Again, this example makes use of only the simplest ISL components, such as near-far, approaching, and movement. The true power of the language will become more evident when we must deal with more complicated environments where schematic concepts such as container or blockage will come into play.

Discussion

In this short paper we have had to elide several important issues that will be the subject of future work – in particular, the origin of schemas, learning of and with schemas, the role of context, and reasoning and inference with schemas.

Our account of image schemas is agnostic about their fundamental origin. Our hunch is that much of image schema structure and function is learned or a result of development. In any case, given some image schema foundations, we do believe new image schemas and their elaborations will be learned, and our goal is to have ISL support this.

ISL provides a language for representing the rich knowledge that we can learn by interacting with the physical world. This representation facilitates the transfer of knowledge learned in one domain to a new, different domain via metaphorical extension. Using semi-Markov decision processes to model the world, a traditional propositional state description would be inflexible; knowledge learned in one domain could not easily be transfered to a new domain. Using ISL, we instead model the structural relationships between objects and their dynamics. We can identify similar structures in new domains, that is, identify the "gist" that captures what we've learned previously about this type of situation. These gists, which could be represented using ISL

state machines as described in the previous section, are essentially learned sequences of image schemas that pertain to particular goals. For example, "catching a cat by sneaking up to it" might be a learned gist. It prescribes a sequence of action schemas given observations of static and dynamic schemas that predictively leads to catching the cat. In addition to learning compositions of image schemas, we may also want to learn specializations of specific image schemas. For example, we might want to learn the difference between "push" and "shove," even though both can be thought of as variations on our "apply force" action schema. This specialization, in turn, helps us better predict outcomes of actions. We are currently working on mechanisms to automate the learning of image schema composition and specialization.

We have touched on a couple of simple examples of reasoning with schemas, such as how to identify that a path is blocked. The example of propagating changes to schema state based on interpretation is tantalizing but requires more work to provide an automated mechanism. In particular, we need to understand the mechanisms for on-the-fly schema combination and interpretation, something humans do with great facility. Some of this may be based on special-case learning, but it may also be the result of general principles. We need to identify these general principles of schema combination so that propagation is well-defined given any novel combination of schemas.

Context plays an important role in interpretation. For example, suppose I have the goal of getting from Los Angeles to San Francisco. As I drive south from my home to the airport, am I "approaching" San Francisco? Not geographically, but if my actions are interpreted as steps in a more abstract plan, then there is a reasonable sense in which the answer is yes. Such context can be represented in schematic terms in ISL as a path over a non-geometrical space, but we have not yet explored the implications of this aspect of metaphorical extension. This is also likely related to issues of attention and goal-directed planning.

Acknowledgments

References

Bailey, D. (1995). Getting a grip: a computational model of the acquisition of verb semantics for hand actions. In *Conference of the International Cognitive Linguistics Association (ICLA 1995)*, Albuquerque, NM.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavior and Brain Sciences* 22: 577-609.

Cohen, P. R. (1998). Maps for verbs. In *Proceedings of the Information and Technology Systems Conference, Fifteenth IFIP World Computer Congress.*

Cohen, P. R., Morrison, C. T. & Cannon, E. (2005). Maps for verbs: The relation between interaction dynamics and verb use. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence (IJCAI 2005).*

Croft, W., & Cruse, D. A. (2004). *Cognitive Linguistics.* Cambridge University Press.

Fikes, R. E. & Kehler, T. (1985). The role of frame-based representation in knowledge representation and reasoning. *Communications of the ACM* 28(9): 904-920.

Fillmore, C. (1968). The case for case. In Bach, E., & Harms, R. T., eds., *Universals in Linguistic Theory.* London: Holt, Rinehart & Winston.

Gentner, D. & Markman, A. M. (1997). Structure mapping in analogy and similarity. *American Psychologist* 52: 45-56.

Gibbs, R. W. & Colston, H. L. (1995). The cognitive psychological reality of image schemas and their transformations. *Cognitive Linguistics* 6(4): 347-378.

Johnson, M. (1987). *The Body in the mind: The bodily basis of meaning, imagination, and reason.* University of Chicago Press, Chicago, IL.

Klein, J. (2002). breve: a 3D simulation environment for the simulation of decentralized systems and artificial life. In *Proceedings of Artificial Life VIII.*

Kosslyn, S. M. (1994). *Image and Brain.* Cambridge, MA: The MIT Press.

Lakoff, G. (1987). *Women, fire and dangerous things.* University of Chicago Press.

Mandler, J. (1992). How to build a baby: ii. Conceptual primitives. *Psychological Review* 99: 597-604.

Mandler, J. (2004). *The foundations of mind: Origins of conceptual thought.* Oxford University Press.

Oakley, T. (2006). Image schema. In D. Geeraerts and H. Cuyckens, editors, *Handbook of cognitive linguistics.* Oxford University Press, *forthcoming.*

Regier, T. (1996). *The human semantic potential: Spatial language and constrained connectionism.* Cambridge, MA: The MIT Press.

Schank, R. C. & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Erlbaum.

SIL International. (2003). LinguaLinks Library. WWW: http://www.sil.orglinguistics/GlossaryOf LinguisticTerms/WhatIsAnImageSchema.htm .

Talmy, L. (2003). *Toward a cognitive semantics - Volume 1: Concept structuring systems (language, speech, and communication).* Cambridge, MA: The MIT Press.